FAA-STD-073
April 16, 2013

U.S. Department
of Transportation

**Federal Aviation
Administration**

NOTE: This draft, dated April 15, 2013, has not been approved and is subject to modification. DO NOT USE PRIOR TO APPROVAL.

U.S. Department of Transportation
Federal Aviation Administration

Standard Practice

PREPARATION OF
JAVA MESSAGING SERVICE DESCRIPTION DOCUMENTS

FOREWORD

This standard is approved for use by all Departments of the Federal Aviation Administration (FAA).

This standard specifies the minimum acceptable content for preparing FAA Java Messaging Service Description Document (JMSDD).

This standard has been prepared in accordance with FAA-STD-068, Department of Transportation Federal Aviation Administration, *Preparation of Standards* [11].

Comments, suggestions, or questions on this document shall be addressed to:

Federal Aviation Administration
Office of Communications, Information and Network Programs, AJM-31
800 Independence Avenue, SW
Washington, DC 20591

Table of Contents

List of Figures

List of Tables

1 SCOPE

This standard provides a set of requirements for developing a Java Messaging Service (JMS) Description Document (JMSDD). The JMSDD provides the details needed to sufficiently describe a JMS-based service as a part of the FAA's implementation of a Service-Oriented Architecture (SOA).

The applicability of this standard is limited to the services that are:

- Designed in accordance with the Java Message Service(JMS) Application Programming Interface (API),
- Supported and enabled by the System Wide Information Management (SWIM) program, and
- Deployed and executed using the NAS Enterprise Message Service (NEMS) as a provider of messaging and infrastructural capabilities.

Any configuration management (CM) or quality assurance (QA) policies, rules or assertions to which the developed JMSDD may be subjected are outside the scope of this standard.

1.1 Introduction

The National Airspace System (NAS) is currently in the process of evolving their traditional mode of information exchange (including highly customized system-to-system interaction) to a paradigm of Service-Oriented Architecture (SOA). The objective of the System Wide Information Management (SWIM) program is to support the development of services in accordance with recognized practices and principles of SOA, and to enable the use of the network and application integration services provided by the FAA Telecommunications Infrastructure (FTI).

In a SOA environment, services are integrated through well-defined interfaces and communicate via messages. SOA, as an architectural style, does not specify any technological solution for implementing services. However, major industry institutions have established a number of open standards for implementing and describing services [32], [26].

In response to increased demand for event-driven, reliable asynchronous message handling and a higher degree of decoupling, many NAS SOA-driven implementations adopted the JMS Application Programming Interface (API) created by Sun Microsystems [JSR-914] as a technological solution for service-oriented development. The JMS API is an open standard for message exchange between loosely-coupled distributed software components by means of Message Oriented Middleware (MOM). The concept of MOM has been realized by multiple vendors in industry as an Enterprise Service Bus (ESB). SWIM has established the NAS implementation of an ESB, termed NAS Enterprise Message Service (NEMS), which provides messaging capabilities and supports integration of all NAS collaborating SOA service nodes.

A service description is an integral and indispensable ingredient of a service deployment. In essence, the service description is an artifact that contains complete human-readable information about "how to interact with the service in order to achieve the required objectives, including the format and content of information exchanged between the service and the consumer and the sequences of information exchange that may be

expected" [24]. All services require a service description. The aim of this standard is therefore to specify requirements for creating a service description document for a JMS-based client according to FAA system engineering standards and practices.

1.2 Intended Audience

The intended audience for this standard includes architects and developers designing, identifying, or developing a system based on the JMS API, and decision makers seeking a better understanding of the application of SOA and Enterprise Messaging principles.

1.3 Basic Concepts

The goal of this section is to establish a clear and unambiguous understanding of several important concepts used in this standard.

1.3.1 Java Message Service Components

JMS is a Java-based application programming interface (API) that provides a common way for Java programs to create, send, receive, and read an enterprise messaging system's messages. [16] A JMS message is created by a message producer and consumed by a message consumer as shown in Figure I.

A JMS-based messaging system is composed of the following parts:

JMS provider – specialized software that accepts messages from sending processes and delivers them to receiving processes (typically across a network). Note: an enterprise messaging product like NEMS, besides being a JMS provider, also provides other capabilities such as fault tolerance, load balancing, mediation support, etc.

Message producer - An application or process that creates and sends messages. A message producer is called a "queue sender" in the point-to-point messaging model and a "topic publisher" in the publish/subscribe messaging model.

Message consumer - An application or process that receives messages. A message consumer is called a "queue receiver" in the point-to-point messaging model and a "topic subscriber" in the publish/subscribe messaging model.

Applications or processes that send and receive messages (*message producers* and *message consumers*) are collectively referred to as *JMS clients*.

Figure I illustrate some of the points described above.

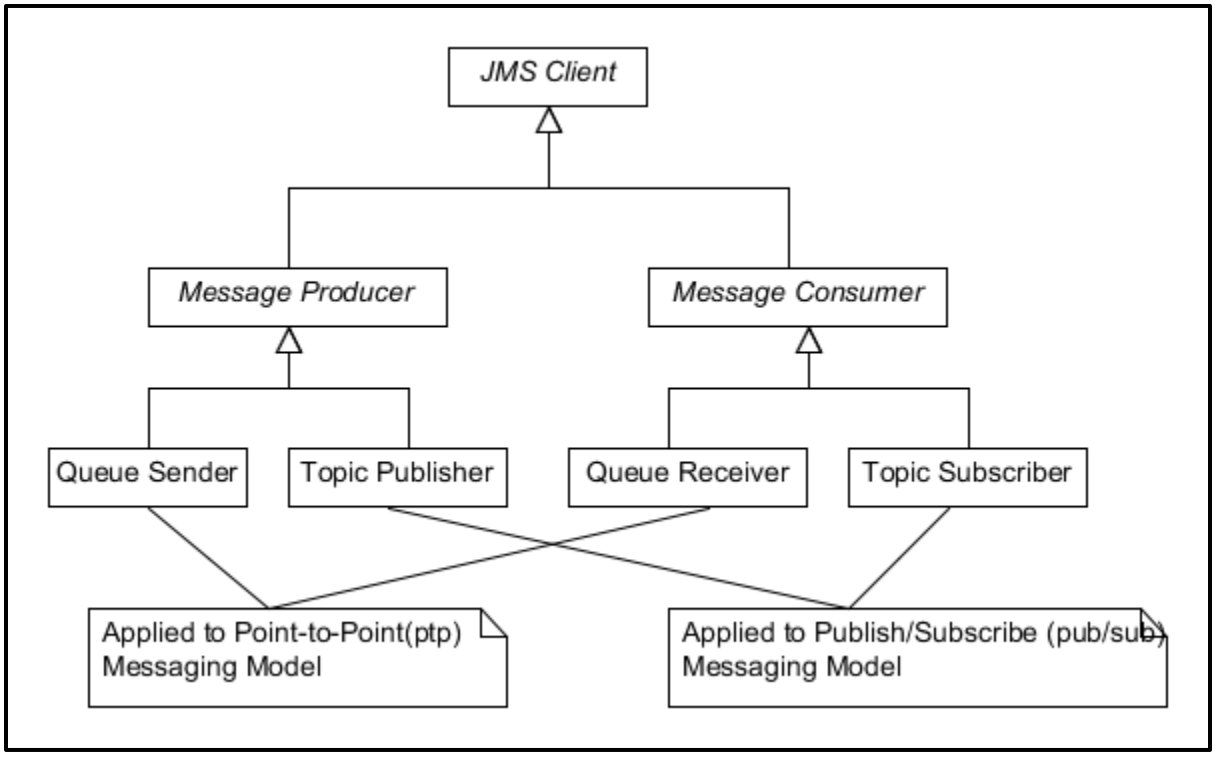

FIGURE . JMS client terminology hierarchy

1.3.2 Messaging Models

The JMS specification supports two messaging models: point-to-point or PTP and publish/subscribe or pub/sub. The following sections describe each of these models.

1.3.2.1 Point-to-Point Messaging Model

The PTP (point-to-point) messaging model is built around the concept of message *queues*, where each queue is a staging area that contains messages that have been sent and are waiting to be read. Each message is addressed to one consumer, which is reminiscent of a one-to-one connection (thus “point-to-point”). The PTP messaging model supports asynchronous “fire and forget” messaging as well as synchronous messaging.

Figure II depicts the PTP messaging model.

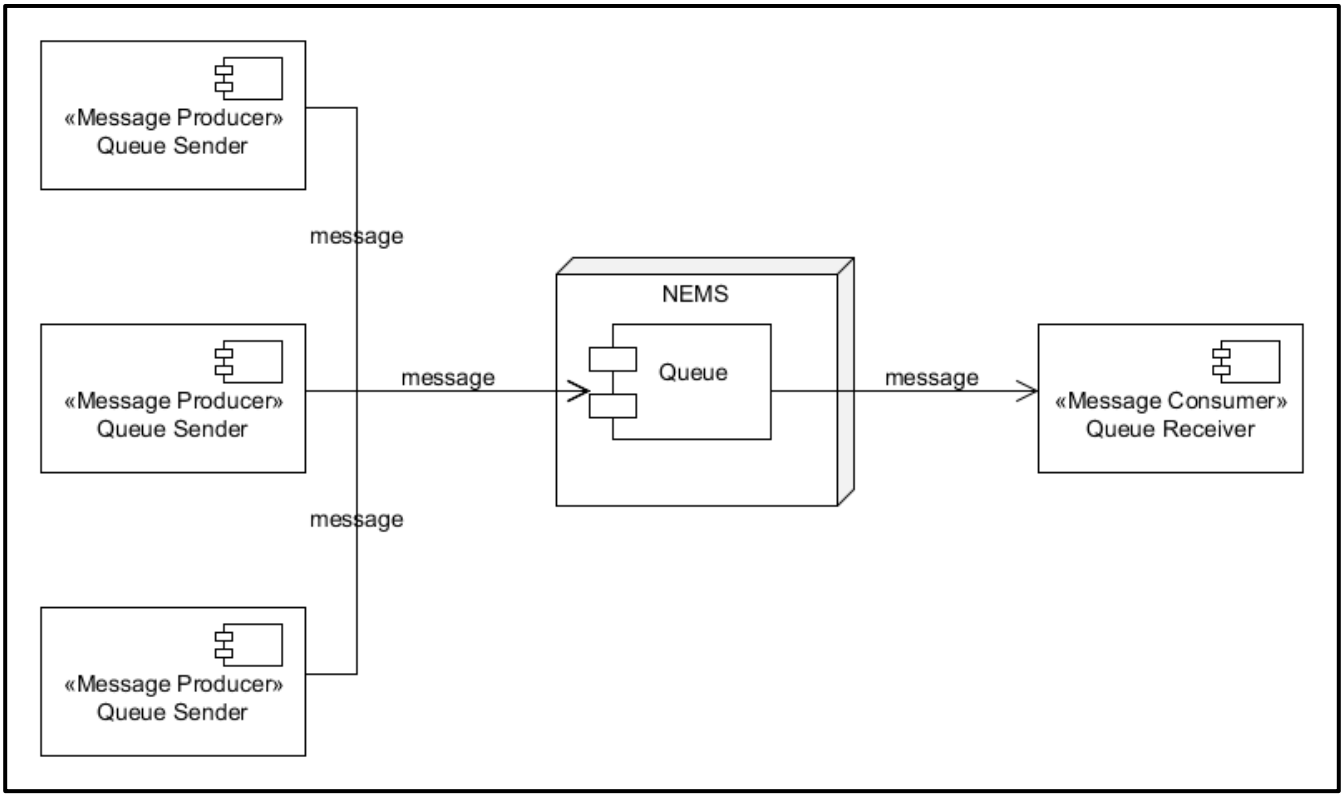

FIGURE . Point-to-Point (PTP) messaging model

1.3.2.2 Publish/Subscribe Messaging Model

The pub/sub messaging model is built around the concept of *topic*, where each message may be delivered to multiple consumers that have subscribed, i.e., registered interest, to a specific topic. The pub/sub messaging model is similar to the notion of one-to-many relationships. The topic is published by a *topic publisher* to a *JMS provider* and then can be subscribed to and consumed by multiple *topic subscribers* (this is sometimes referred to as broadcasting the messages). Thus a *topic* is a distribution mechanism for publishing messages that are delivered to multiple subscribers.[41]

Figure III depicts the pub/sub messaging model.

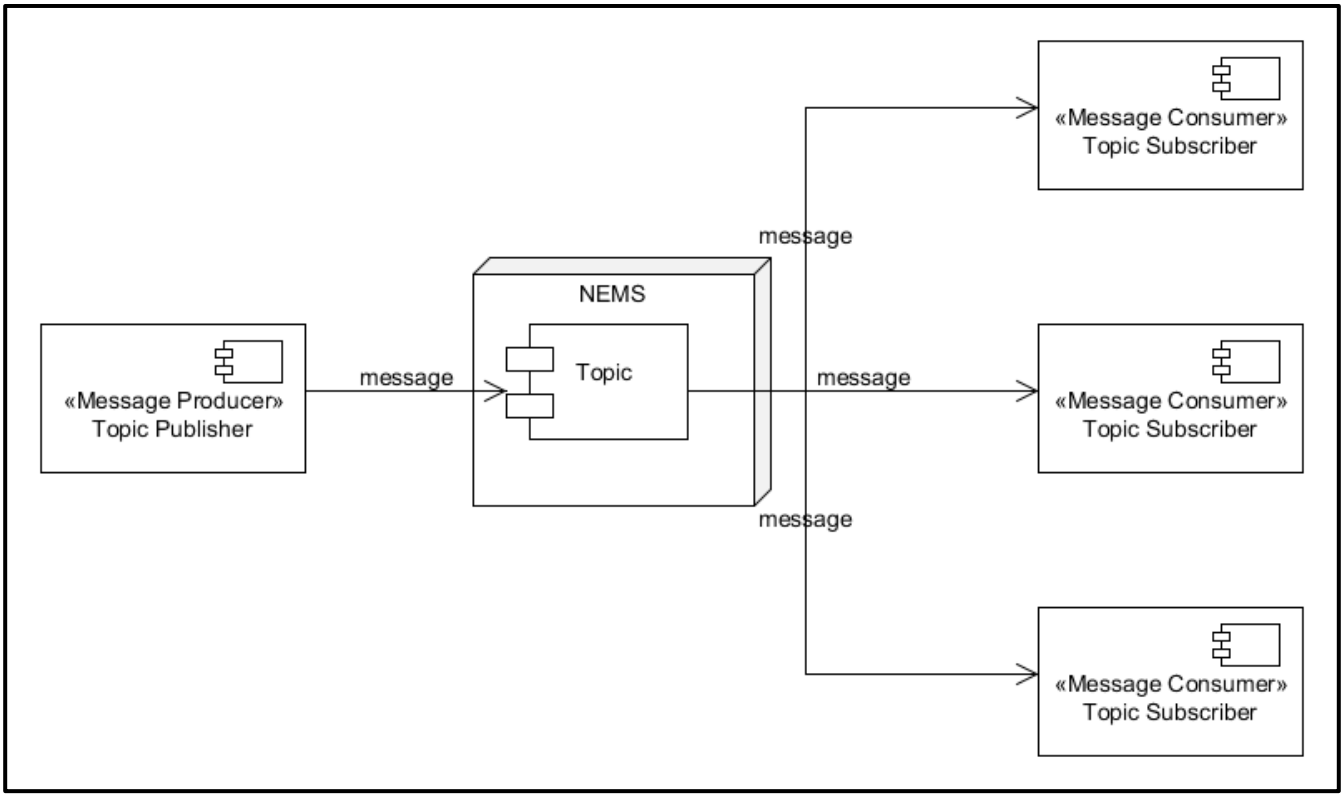

FIGURE . Publish/subscribe messaging model

1.3.3 JMS Message

In a JMS-based messaging system, a *message* is a self-contained package of business data and routing headers. JMS messages are entities that are used to exchange information between JMS clients.

The message's header and property information is used by a JMS provider for routing, filtering, and delivering the payload. The payload is the content of the message.

A JMS message consists of three parts:

Header – all messages support a standard set of header fields. Header fields contain values used by both clients and providers to identify and route messages. [43]

Properties – in addition to the standard header fields, messages provide a built-in facility for adding optional header fields to a message, e.g., properties are used to expose data for message filtering by a JMS provider. JMS defines three types of properties: JMS-defined, provider-specific, and application-specific.

- *JMS-defined properties* are properties established by the JMS specification. The difference between JMS-defined properties and JMS headers is that vendors can choose not to support JMS-defined properties, or to support some or all of them.

- *Provider-specific properties* are properties that can be established by JMS provider to support proprietary vendor features.
- *Application-specific properties* are properties that are defined by the application developer.

Body (or Payload) – the actual (business) data transferred by the message. JMS defines several types of message body which cover the majority of messaging styles currently in use. [43]

Figure IV depicts the architecture of a JMS message.

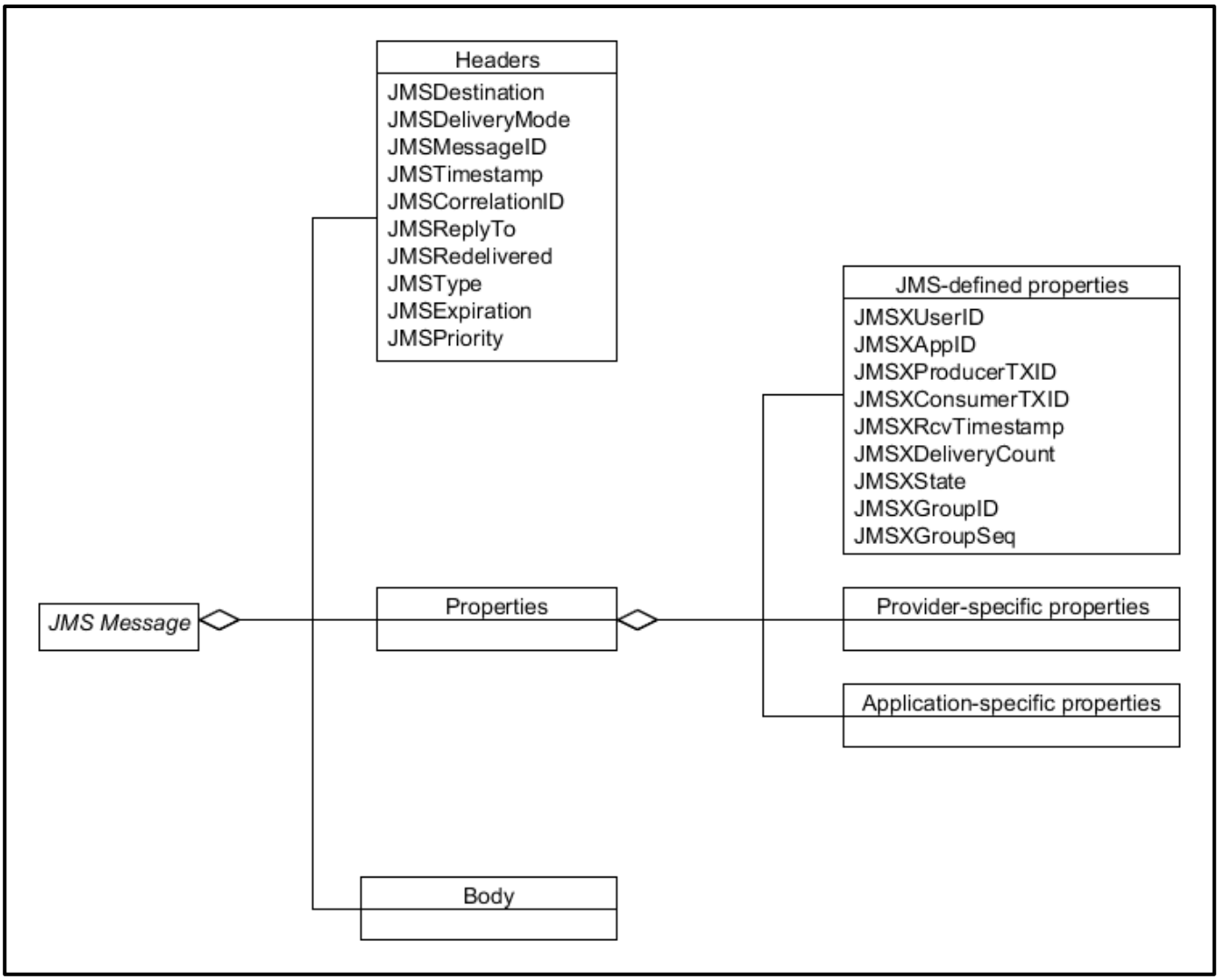

FIGURE . JMS message architecture

2 APPLICABLE DOCUMENTS

2.1 Government Documents

[1] FAA Order 1000.36, FAA Writing Standards, 31 March 2003.
http://www.faa.gov/documentlibrary/media/order/branding_writing/order1000_36.pdf

[2] FAA Order 1700.6C, FAA Branding Policy, Use of the FAA Logo, FAA Signature, and DOT Seal, 11 September 2006.
http://www.faa.gov/documentLibrary/media/order/branding_writing/Branding_Order_17006.pdf

[3] Federal Data Registry, Federal Aviation Administration Section
https://fdr.gov/fdr/Home.jsp

[4] FAA-HDBK-008, Using FAA Standards to Describe and Register Web Services, 4 February 2011.
http://www.tc.faa.gov/its/worldpac/standards/faa-hdbk-008.pdf

[5] FAA-STD-025f, Preparation of Interface Documentation, 30 November 2007.
http://www.tc.faa.gov/its/worldpac/standards/faa-std-025f.pdf

[6] FAA-STD-060b, Data Standard for the National Airspace System, 8 March 2004.
http://www.tc.faa.gov/its/worldpac/standards/faa-std-060b.pdf

[7] FAA-STD-063, XML Namespaces, 1 May 2009.
http://www.tc.faa.gov/its/worldpac/standards/faa-std-063.pdf

[8] FAA-STD-064, Web Service Registration, 1 May 2009.
http://www.tc.faa.gov/its/worldpac/standards/faa-std-064.pdf

[9] FAA-STD-065, Web Service Description Document, 26 February 2010
http://www.tc.faa.gov/its/worldpac/standards/faa-std-065.pdf

[10] FAA-STD-066, Web Service Taxonomies, 26 February 2010.
http://www.tc.faa.gov/its/worldpac/standards/faa-std-066.pdf

[11] FAA-STD-068, Preparation of Standards, 4 December 2009.
http://www.tc.faa.gov/its/worldpac/standards/faa-std-068.pdf

[12] FAA-STD-070, Preparation of Web Service Requirements Documents, 12 July 2012.
http://www.tc.faa.gov/its/worldpac/standards/faa-std-070.pdf

[13] Federal Standard 1037C, Glossary of Telecommunication Terms, 7 August 1996.
http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm

[14] NAS System Engineering Manual, Version 3.1, 6 June 2006.
http://fast.faa.gov/SystemEngineering.cfm

[15] U.S. Government Printing Office Style Manual, 30th edition 2008.
http://www.gpoaccess.gov/stylemanual/

[16] SWIM Controlled Vocabulary, March 2013
http://www.faa.gov/go/swimvocabulary

2.2 Non-Government Documents

[17] DCMI Glossary, Dublin Core Metadata Initiative, User Guide Committee, 23 April 2004.
http://dublincore.org/documents/usageguide/glossary.shtml

[18] Glossary of Security Terms, SANS Institute, August 2011.
http://www.sans.org/resources/glossary.php

[19] ISO/IEC 11179-1, Information Technology – Metadata Registries (MDR) – Part 1: Framework, 15 September 2004.
http://metadata-standards.org/11179/

[20] ISO/IEC 11404, Information technology — General-Purpose Datatypes (GPD), Second Edition, 15 December 2007.
http://standards.iso.org/ittf/PubliclyAvailableStandards/c039479_ISO_IEC_11404_2007(E).zip

[21] ISO/IEC 12207: 2008 Information Technology – Software Life Cycle Processes, 2008.
http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=43447

[22] ISO/IEC 6523-1, Structure for the Identification of Organizations and Organization Parts, 1998.
http://www.iso.org/iso/catalogue_detail?csnumber=25773

[23] ISO/IEC CD 20944-002, Information Technology – Metadata Interoperability and Bindings (MDIB) – Part 002, Common Vocabulary, 12 April 2004.
http://jtc1sc32.org/doc/N1101-1150/32N1105T-CD20944-002.pdf

[24] OASIS Reference Model for SOA 1.0, 12 October 2006.
http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

[25] OASIS Reference Ontology for Semantic Service Oriented Architectures, Public Review 1, 5 November 2008.
http://www.oasis-open.org/apps/group_public/download.php/29909/Reference Ontology for Semantic Service Oriented Architectures_Public_Review_1.doc

[26] OGC Web Services Common Standard, Version 2.0.0, Open Geospatial Consortium Inc., 7 April 2007.
http://www.opengeospatial.org/standards/common

[27] Oracle Java Message Specification, Version 1.1, April 12, 2002
http://www.oracle.com/technetwork/java/jms/index.html

[28] OWL-S: Semantic Markup for Web Services, W3C Member Submission, 22 November 2004.
http://www.w3.org/Submission/OWL-S

[29] RFC 2119, Key words for Use in RFCs to Indicate Requirement Levels, Network Working Group, March 1997.
http://www.rfc-editor.org/rfc/rfc2119.txt

[30] RFC 2828, Internet Security Glossary, Network Working Group, May 2000.
http://www.ietf.org/rfc/rfc2828.txt

[31] RFC 3986, Uniform Resource Identifier (URI): Generic Syntax, Network Working Group, January 2005.
http://www.rfc-editor.org/rfc/rfc3986.txt

[32] Web Services Architecture, W3C Working Group Note, 11 February 2004.
http://www.w3.org/TR/ws-arch

[33] Web Services Description Language (WSDL) 1.1, W3C Note, 15 March 2001.
http://www.w3.org/TR/wsdl

[34] Web Services Description Language (WSDL) Version 2.0 Part 0: Primer, W3C Recommendation, 26 June 2007.
http://www.w3.org/TR/wsdl20-primer/

[35] Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Recommendation, 26 June 2007.
http://www.w3.org/TR/wsdl20/

[36] Web Services Description Requirements, W3C Working Draft, 28 October 2002.
http://www.w3.org/TR/2002/WD-ws-desc-reqs-20021028/

[37] Web Services Glossary, W3C Working Draft, 14 November 2002.
http://www.w3.org/TR/2002/WD-ws-gloss-20021114/

[38] Web Services Policy 1.5 – Framework, W3C Recommendation, 04 September 2007.
http://www.w3.org/TR/2007/REC-ws-policy-20070904/

[39] XML Schema Part 1: Structures Second Edition, W3C Recommendation, 28 October 2004.
http://www.w3.org/TR/xmlschema-1/

[40] XML Schema Part 2: Datatypes Second Edition, W3C Recommendation, 28 October 2004.
http://www.w3.org/TR/xmlschema-2/

[41] Wikipedia
http://en.wikipedia.org/wiki/

[42] Java 2 Platform, Enterprise Edition, v 1.3 API Specification
http://docs.oracle.com/javaee/1.3/api/

[43] Java Message Service Specification Version 1.1: Sun Microsystems, Inc.: April 12, 2002
http://download.oracle.com/otn-pub/jcp/7195-jms-1.1-fr-spec-oth-JSpec/jms-1_1-fr-spec.pdf?AuthParam=1364588309_98fc713e75f0a6a10ac79989907c46c2

2.3 Order of Precedence

In the event of a conflict between the text of this document and the references cited herein, the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

3 DEFINITIONS

3.1 Key Words

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", “MAY", and "OPTIONAL" in this standard are to be interpreted as described in RFC 2119 [29]. These key words are capitalized when used to unambiguously specify requirements. When these words are not capitalized, they are meant in their natural-language sense.

All examples in the document are labeled as "non-normative", which means they are not to provide a canonical implementation, but merely to illustrate technical features of a particular approach.

3.2 Terms and Definitions

Asynchronous	An interaction in which the associated messages are chronologically and procedurally decoupled. For example, in a request-response interaction, the client agent can process the response at some indeterminate point in the future when its existence is discovered.[32]
Audit	A process that records information needed to establish accountability for system events and for the actions of system entities that cause them. [30]
Authentication	The process of verifying an identity claimed by or for a system entity. [30]
Authorization	The granting of rights or permission to a system entity (mainly but not always a user or a group of users) to access a service. [12]
Binding	An association between an interface, a concrete protocol, and a data format. A binding specifies the protocol and data format to be used in transmitting messages defined by the associated interface. [36]
Business Function	A characteristic action or activity that needs to be performed to achieve a desired objective, or in the context of this standard, to achieve a real world effect. (Adapted from [14])
Business Service	A business function or capability that is offered as a service.[16]
Confidentiality	Protective measures that assure that information is not made available or disclosed to unauthorized individuals, entities, or processes (i.e., to any unauthorized system entity). [30]
Data Element	A unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes.
Datatype	A set of distinct values, characterized by properties of those values, and by operations on those values.[20]

Effect	A state or condition that results from interaction with a service. Multiple states may result depending on the extent to which the interaction completes successfully or generates a fault.
End Point	An association between a fully-specified binding and a physical point (i.e., a network address) at which a service may be accessed.
Fault	A message that is returned as a result of an error that prevents a service from implementing a required function. A fault usually contains information about the cause of the error. [12]
Format	The arrangement of bits or characters within a group, such as a data element, message, or language. [12]
Idempotent	A term used to describe an operation in which a given message will have the same effect whether it is received once or multiple times; i.e., receiving duplicates of a given message will not cause any undesirable effect. [12]
Input	Data entered into, or the process of entering data into, an information processing system or any of its parts for storage or processing. (Adapted from [23])
Integrity	Protective measures that assure that data has not been changed, destroyed, or lost in an unauthorized or accidental manner. [30]
Java Message Service (JMS)	A Java-based application programming interface (API) that provides a common way for Java programs to create, send, receive, and read an enterprise messaging system's messages.[16]
JMS Client	An application or process that produces and/or receives messages. [41]
JMS Provider	A messaging system that implements JMS in addition to the other administrative and control functionality required of a full featured messaging product.[42]
Message	A basic unit of communication from one software agent to another sent in a single logical transmission.[16]
Message Consumer	A JMS client that receives messages.[41]
Message-Oriented Middleware (MoM)	Software or hardware infrastructure supporting sending and receiving messages between distributed systems.[16]
Message Producer	A JMS client that creates and sends messages.
Namespace	A collection of names, identified by a URI reference, that are used in XML documents as element types and attribute names. The use of XML namespaces to uniquely identify metadata terms allows those terms to be unambiguously used across applications, promoting the possibility of shared semantics. [17]

NAS Enterprise Message Service	A NAS-based implementation of message-oriented middleware (MOM) that is responsible for distributing messages among information consumers and providers, as well as providing administrative functionality that includes (but is not limited to) fault tolerance, load balancing, mediation and orchestration support. [16]
Non-Repudiation	Protective measures against false denial of involvement in a communication. [30]
Normative Document	A document that provides rules, guidelines, or characteristics for activities or their results. Note: The term "normative document" is a generic term that covers such documents as standards, technical specifications, codes of practice, and regulations. [23] In the context of this standard, a normative document is a set of rules that (1) determines the behavior of interacting entities and (2) has been developed by a recognized body in industry or academia and established by consensus in the FAA.[12]
Operation	A set of messages related to a single service action. (Adapted from [36])
Organization	A unique framework of authority within which a person or persons act, or are designated to act, towards some purpose. Any department, service, or other entity within an organization which needs to be identified for information exchange. [16]
Output	Data transferred out of, or the process by which an information processing system or any of its parts transfers data out of, that system or part. (Adapted from [23])
Permissible Values	The set of allowable instances of a data element. [12]
Precondition	A state or condition that is required to be true before an action can be successfully invoked.
Protocol	A formal set of conventions governing the format and control of interaction among communicating functional units. [16]
Quality of Service (QoS)	A parameter that specifies and measures the value of a provided service.[16]
Queue Receiver	A JMS Client that receives messages from a queue.[16]
Queue Sender	A JMS Client that sends a message to a queue.[16]
Real World Effect	An ultimate purpose associated with the interaction with a particular service. It may be the response to a request for information or the change in the state of some entities shared between the participants in the interaction.[16]
Registry	An enabling infrastructure that uses a formal registration process to store, catalog, and manage metadata relevant to a service. A registry supports the search, identification, and understanding of resources, as well as query capabilities. [8]

Security	The protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them. [21]
Security Mechanism	A process (or a device incorporating such a process) that can be used in a system to implement a security service that is provided by or within the system. [30]
Service	A mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description.[16]
Service Consumer	An organization that seeks to satisfy a particular need through the use of capabilities offered by means of a service. [16]
Service Description	The information needed in order to use, or consider using, a service.[16]
Service Interface	The means by which the underlying capabilities of a service are accessed.[16]
Service Provider	An organization that offers the use of capabilities by means of a service.[16]
Service-Oriented Architecture (SOA)	A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. A SOA provides a uniform means to offer, discover, interact with, and use capabilities to produce desired effects consistent with measurable preconditions and expectations.[16]
Structured Data	Data that is organized in well-defined semantic “chunks” or units that are variously called fields, elements, objects, or entities. Individual units are often combined to form larger, more complex units. (Adapted from [5])
Synchronous	An interaction in which the participating agents must be available to receive and process the associated messages from the time the interaction is initiated until all messages are actually received or some failure condition is determined.[32]
Taxonomy	A system or controlled list of values by which to categorize or classify objects. [10]
Topic	A distribution mechanism for publishing messages that are delivered to multiple subscribers.
Topic Publisher	A JMS client that sends messages to a topic. [16]
Topic Subscriber	A JMS client that retrieves messages from a topic. [16]
Uniform Resource Identifier (URI)	A compact string of characters for identifying an abstract or physical resource. [31]
Uniform Resource Locator (URL)	A type of URI that identifies a resource via a representation of its primary access mechanism (e.g., its network "location"), rather than by some other attributes it may have. [31]

Unstructured Data	Data that does not follow any hierarchical sequence or any relational rules. Examples of unstructured data may include audio, video, and unstructured text such as the body of an e-mail or word processor document. (Adapted from [5])
User	A human, his/her agent, a surrogate, or an entity that interacts with information processing systems. [23] A person, organization entity, or automated process that accesses a system, whether authorized to do so or not. [18]
Web Service	A platform-independent, loosely-coupled software component designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format. Other systems interact with the Web service in a manner prescribed by its description by means of XML-based messages conveyed using Internet transport protocols in conjunction with other Web-related standards. [16]

3.3 Acronyms and Abbreviations

AIM	Aeronautical Information Management
AIXM	Aeronautical Information Exchange Model
API	Application Programming Interface
DCMI	Dublin Core Metadata Initiative
DOT	Department of Transportation
EDD	External Data Document
ESB	Enterprise Service Bus
FAA	Federal Aviation Administration
FDR	FAA Data Registry
FTI	FAA Telecommunications Infrastructure
FTP	File Transfer Protocol
GML	Geography Markup Language
HTTP(S)	Hypertext Transfer Protocol (Secure)
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
JMS	Java Message Service
JMSDD	Java Messaging Service Description Document

JNDI	Java Naming and Directory Interface
MDR	Metadata Registry
MEP	Message Exchange Pattern
MOM	Message-Oriented Middleware
MTBCF	Mean Time Between Critical Failure
MTBF	Mean Time Between Failure
MTTR	Mean Time To Restore
NAS	National Airspace System
NEMS	NAS Enterprise Message Service
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OWL-S	Ontology Web Language (OWL)-based Web Service Ontology
PNG	Portable Network Graphics
PTP	Point-to-Point
QoS	Quality of Service
RFC	Request For Comment
RPC	Remote Procedure Call
SOA	Service-Oriented Architecture
SOAP	Originally “Simple Object Access Protocol”; the full spelling is no longer used
SVG	Scalable Vector Graphics
SWIM	System Wide Information Management
TCP/IP	Transmission Control Protocol/Internet Protocol
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WS	Web Service
WSDD	Web Service Description Document
WSDL	Web Service Description Language

WSDOM	Web Service Description Ontological Model
XML	eXtensible Markup Language

4 GENERAL REQUIREMENTS

This section describes requirements for the stylistic aspects of the JMSDD. Detailed requirements for the structure and content of the JMSDD are provided in section 5.

4.1 Text, Grammar and Style

a. The text SHALL be written in clear and simple language, free of vague terms, or those subject to misinterpretation.

b. All sentences SHOULD be complete and grammatically correct. Refer to FAA Order 1000.36, FAA Writing Standards [1] for guidance.

c. The United States Government Printing Office Style Manual [15] SHALL be used as a guide for capitalization, spelling, punctuation, syllabification, compounding words, tabular work, and other elements of grammar and style.

4.2 Page Numbering

a. The front cover page SHALL omit the page number.

b. All pages after the front cover page and before the page containing the first (“Scope”) section SHALL be numbered consecutively with lower-case Roman numerals, starting with ii (for example, ii, iii, and iv).

c. The first page of the first (“Scope”) section SHALL be numbered with an Arabic numeral 1.

d. All subsequent pages SHALL be numbered sequentially using Arabic numerals.

e. The page numbers SHALL be placed at the bottom center of each page.

4.3 Page Headers

a. Each page, including the front cover, SHALL contain a header in the upper right-hand corner right-justified.

b. Each header SHALL contain the JMSDD Identifier. Note: In most cases, the identifier is assigned by a governing or configuration management organization under whose authority the service is developed or functions.

c. If the JMSDD is a revision to a baseline JMSDD, the word “Revision” followed by the revision letter SHALL be included immediately under the JMSDD Identifier.

d. If the JMSDD has been approved by a governing or configuration management control organization, the header SHALL include the date of JMSDD approval on the last line.

e. If the JMSDD is a draft, the header SHALL include the word “DRAFT” in capital letters under the JMSDD identifier and the date of the draft on the next line.

4.4 Use of Hyperlinks

To improve the readability and understanding of the JMSDD, usage of hyperlinks is prescribed as follows:

a. Every term that is used and defined within the JMSDD SHOULD be linked via a hyperlink reference to the location of its definition in the JMSDD's "Definitions" section.

b. When the same term is used more than once within the same sentence or paragraph, only the first occurrence of the term SHOULD be referenced.

c. Every document that is cited within the JMSDD SHALL be linked via a hyperlink reference to the location of its bibliographic entry in the JMSDD's "Applicable Documents" section.

d. When a document is quoted within the JMSDD, the quote SHALL include a hyperlink reference to the location of the document's bibliographic entry in the JMSDD's "Applicable Documents" section.

4.5 Use of Diagrams

There are a number of sections in the JMSDD where using diagrams is suggested to enhance the understanding of a described topic.

a. Unified Modeling Language (UML) diagrams are RECOMMENDED since UML is able to concisely describe concepts without implying any specific technology. Information about UML diagrams is available at http://www.uml.org/.

4.6 Identifying Figures, Tables, and Appendices

a. Figures SHALL be identified by "Figure", the level one section number in which they appear followed by a dash and numbered sequentially using Arabic numerals within the level one section (e.g., 3-1, 3-2, 3-3, 4-1, 4-2), followed by the figure title.

b. The figure identification SHALL be placed below the figure.

c. Tables SHALL be identified by "Table", the level one section number in which they appear followed by a dash and numbered sequentially using Arabic numerals within the level one section (e.g., 3-1, 3-2, 3-3, 4-1, 4-2), followed by the table title.

d. The table identification SHALL be placed above the table.

e. Appendices SHALL be identified by "Appendix", followed by sequential capital letters (e.g., Appendix A), and the appendix title.

5 DETAILED REQUIREMENTS

This section describes requirements for the structure and content of the JMSDD.

5.1 Cover Page

a. The JMSDD SHALL include a cover page as the first page.

b. The upper left corner of the cover page SHALL include the FAA signature (the Department of Transportation triskelion figure with the words "U.S. Department of Transportation" and the words "Federal Aviation Administration" below it) in accordance with FAA Order 1700.6, FAA Branding Policy [2].

c. The line "Java Messaging Service Description Document" SHALL be centered above the title.

d. The title SHALL be the name by which the service will be known. Note: In most cases, the title will consist of the approved service's name issued by the activity authorized to assign the name. That name will be referred to throughout the JMSDD as the service name.

An example of a JMSDD cover page is shown in Appendix A.

5.2 Approval Page (Optional)

Signatures on this page ensure that the interested parties have approved the JMSDD content. The approval page may not be required based on the configuration management policies established within a given organization. The following statements apply when signed approval is required.

a. The approval page SHALL be the first interior page of the JMSDD.

b. The approval page SHALL contain the line "Java Messaging Service Description Document" centered above the title of the service, and the line "Approval Signatures" centered below the title of the service.

c. The approval page SHALL include information for every cosigner.

d. The information SHALL include the cosigner's full name.

e. The information SHALL include the full name of the cosigner's organization followed by the acronym by which the organization is commonly recognized within FAA.

f. The information SHALL include the cosigner's signature.

g. The information SHALL include the date of the signature.

An example of a JMSDD Approval Page is shown in Appendix B.

5.3 Revision Record Page

a. The JMSDD SHALL include a revision record page.

b. The revision record page SHALL contain the centered line "Revision Record" above the revision record table.

c. Only revisions SHALL be listed.

d. The revision record page SHALL include information for every revision listed.

e. The information SHALL include the revision letter.

f. The information SHALL include a brief description of the revision.

g. The information SHALL include the date of the revision.

h. The information SHALL include the full name of the person who entered this revision record ("Entered by").

An example of a JMSDD Revision Record Page is shown in Appendix C.

5.4 Table of Contents

a. The JMSDD SHALL include a table of contents.

b. The JMSDD SHALL conform to the basic outline shown in Table I below. Note: the sections shown in italics are optional.

TABLE . JMSDD table of contents

5.5 Scope

a. Section 1 of the JMSDD SHALL provide a scope statement that is a clear, concise abstract of the coverage of the JMSDD.

b. Section 1 of the JMSDD MAY include paragraphs on the service's purpose, applicability, background, etc. as needed to give readers of the JMSDD a context for understanding the body of the JMSDD.

5.6 Applicable Documents

a. Section 2 of the JMSDD SHALL list all documents specifically cited in the JMSDD.

b. Only documents that are specifically cited in the JMSDD SHALL be listed in section 2.

c. Section 2 of the JMSDD SHALL present bibliographic information about each document listed.

d. The information SHALL include the full title of the document.

e. The information SHOULD include the alternate title or abbreviated name by which the document is known or recognized.

f. The information SHALL include the publisher of the document.

g. The information SHALL include the publication date of the document.

h. The information SHOULD include the appropriate version of the document (e.g., the latest version, the version needed for compatibility with other documents, the version of the document that is under contract by the project.)

i. The information SHOULD include the creator of the document, if it is different from the publisher.

j. The information SHALL include the address or location (preferably a persistent Web location, i.e., URL) where a copy of the document can be obtained.

k. Section 2.1 of the JMSDD SHALL list all types of Government standards and other publications cited in the JMSDD.

l. Section 2.2 of the JMSDD SHALL list all types of non-Government standards and other publications cited in the JMSDD.

m. When a document is quoted within the JMSDD, the quote SHOULD indicate where in the document the quote is to be found (e.g., using section number, paragraph number, page number, or other means of identification).

5.7 Definitions

a. Section 3.1 of the JMSDD SHALL define all terms used in the JMSDD to provide for clarity, unless the terminology is generally accepted and not subject to misinterpretation.

b. Only terms that are specifically used in the JMSDD SHALL be listed in section 3.1.

c. Definitions MAY be included by reference to another document.

d. Terms and their definitions SHALL be listed in alphabetical order.

e. Section 3.2 of the JMSDD SHALL include a list of acronyms and abbreviations used in the JMSDD, together with their full spelling.

f. Only acronyms and abbreviations that are specifically used in the JMSDD SHALL be listed in section 3.2.

g. Acronyms and abbreviations SHALL be listed in alphabetical order.

5.8 Service Profile

Section 4 of the JMSDD identifies and describes the business service, its provider, its known consumers, its functional and non-functional characteristics, and constraints over its capabilities.

a. Section 4 of the JMSDD SHALL present information about the service profile.

b. The information SHALL include the name of the service.

c. The name SHALL be identical with the name of the service provided on the cover page of the JMSDD.

d. The information SHALL include a service unique identifier, i.e., URI, as established by the FAA registration authority (3).

e. The information SHALL include a description of the service.

f. The information SHALL include a service version or revision level.

g. The information SHALL include a service category.

h. One or more values representing the service category SHALL be selected from the FAA Service Category Taxonomy described in section 5.3.5 of FAA-STD-066, Web Service Taxonomies. [10].

i. The information SHALL include the lifecycle stage of the service.

j. The single value representing the service's lifecycle stage SHALL be selected from the Lifecycle Stage Taxonomy described in section 5.3.7 of FAA-STD-066, Web Service Taxonomies. [10].

k. The information SHALL include the level of criticality for the service.

l. The single value representing the service's criticality level SHALL be selected from the Service Criticality Taxonomy described in section 5.3.8 of FAA-STD-066, Web Service Taxonomies. [10].

m. The information MAY include additional service classifications using business domain-specific taxonomies.

5.8.1 Service Provider

This standard treats the service provider as an organization responsible for establishing and maintaining the business service.

a. Section 4.1 of the JMSDD SHALL present information about the provider organization.

b. The information SHALL include the name of the organization.

c. The provided name SHALL consist of the full name spelled out followed by the acronym by which it is commonly recognized within FAA.

d. The information SHALL include an organization unique identifier, i.e., URI, as established by the FAA registration authority (FDR).

e. The information MAY include a brief description of the organization.

f. The information MAY include a URL for the organization's Web page.

5.8.1.1 Point of Contact

a. Section 4.1.1 of the JMSDD SHALL present information for a point of contact, i.e., a person or group within the provider organization, suitable for making a human contact for any purpose.

b. The information SHALL include the full name of the contact.

c. The information SHALL include the contact's job title or a brief description of the contact's responsibilities.

d. The information SHALL include at least one telephone number.

e. The information SHALL include at least one e-mail address.

f. The information MAY include a postal address.

g. If the required point of contact information is maintained at a persistent Web location (URL), section 4.1.1 of the JMSDD MAY include a hyperlink to this location in lieu of including the information itself.

5.8.2 Service Consumers

a. Section 4.2 of the JMSDD SHALL present information about each known service consumer.

b. The information SHALL include the consumer organization's full name and acronym.

c. The information SHOULD include an accessible reference (e.g., URL) for the Web page that supplies information about the organization.

5.8.3 Service Functionality

This standard asserts that every service represents a set of one or more identifiable business functions. The goal of section 4.3 of the JMSDD is to describe the business function(s) and the real world effects that result from invoking these business functions from a business point of view, that is, from the point of view of consumer organizations that will use the service to conduct their business. Section 4.3 should not address the mechanics of invocation (this aspect is addressed in the Service Interface section), but rather it should focus on answering the question of what the service does and what is the ultimate result of using the service. The ultimate result of using a service is referred to as the "real world effect". The real world effect may include:

"1. Information returned in response to a request for that information,
2. A change to the shared state of defined entities, or
3. Some combination of (1) and (2)." [24]

For example, a real world effect could be knowledge that "the flight has been rerouted" (change in the state) or a "weather forecast" (response to a request for information).

a. Section 4. 3 of the JMSDD SHALL present information about the service functionality from the business perspective.

b. The information SHALL describe the service's business function(s) and real world effect(s).

c. The service's business function(s) SHALL be correlated with the real world effect(s).

5.8.4 Security

This standard defines service security as collective measures that enable the service to provide protection against security threats. These threats may include (but are not limited to): unauthorized access to service information; unauthorized disclosure, modification and destruction of information; unknown status and repudiation in execution; and denial of service. To address the security threats, security mechanisms are utilized. Table II presents a list of the most typical security mechanisms commonly implemented by JMS applications, together with their intended purpose. Note: this list is neither exhaustive nor prescriptive.

TABLE . Java message service security mechanisms

Mechanism	*Purpose*
Authentication	To assure that system entities (individuals, entities, or processes) are who they claim to be.
Authorization	To assure that system entities have been granted the right or permission to access a service.
Integrity	To assure that data has not been changed, destroyed, or lost in an unauthorized or accidental manner.
Confidentiality	To assure that information is not made available or disclosed to unauthorized system entities.
Non-Repudiation	To assure that the sender or recipient of a message cannot legitimately claim that they did or did not participate in the message exchange.
Audit	To record information needed to establish accountability for system events and for the actions of system entities that cause them.

a. Section 4.4 of the JMSDD SHALL present information about the Java message service security.

b. Section 4.4 of the JMSDD SHALL describe all security mechanisms implemented by the JMS application.

c. Section 4.4 of the JMSDD SHALL specify all the security zones where the service is to be deployed, indicating whether it is provisioned in the "trusted" or "untrusted" regions. Note: provisioning in the untrusted region enables connectivity between NAS and Non-NAS for access to the service.

d. When a security mechanism is implemented by using a standard protocol or specification document, the information about this document SHALL be presented as prescribed in section 5.6 of this standard.

e. When the JMS client delegates one or more security measures to an external security service, the document that specifies the external security service SHALL be presented as prescribed in section 5.6 of this standard.

5.8.5 Qualities of Service

This standard defines Qualities of Service (QoS) as measurable characteristics that the service is expected to meet or possess. To be usable in practice, these QoS should be documented in a way that ensures clear and common understanding for the service stakeholders. Section 4.5 of the JMSDD lists all QoS parameters associated with the JMS client that produces messages.

a. Section 4.5 of the JMSDD SHALL list all Quality of Service (QoS) parameters associated with the provided service.

b. Section 4.5 of the JMSDD SHALL include the values for QoS parameters as defined in table 3. Note: These parameters are required by NEMS on-ramping process.

c. Section 4.5 of the JMSDD SHALL present information about each QoS parameter listed.

d. When QoS parameters, other than listed in table III, are identified, section 4.5 of the JMSDD SHALL present information about each QoS parameter. Note:

e. The information SHALL include the QoS parameter's name.

f. The information SHALL include the QoS parameter's value or range of values.

g. The information SHALL include the QoS parameter's definition. See section 6.2 of FAA-STD-064 [8] to help ensure that the definition is as informative and understandable as possible.

h. The information SHALL include a description of how the values are measured or calculated.

i. The information SHALL include the unit of measure (e.g., seconds, percentage).

Appendix D contains examples of additional user-defined QoS parameters as they might be presented in the JMSDD.

Table . Required QoS parameters

QoS Parameter Name	Definition	Method	Unit of Measure
Availability	A measure of the lowest probability that a system or constituent piece will be operational during any randomly selected period of time, or, alternatively, the fraction of the total available operating time that the system or constituent piece is operational. (FAA SEM 4.8.2 [11])	100 * ((24 – Total Outage Time in Hours) / 24). Measurements are taken daily and apply to the preceding 24-hour period.	Percentage, accurate to 3 decimal places.

Mean Time Between Failure (MTBF)	Average time between hardware or software component failures that do not result in the loss of the service.	The sum of the individual times between noncritical failures divided by the number of noncritical failures.	Hours.
Mean Time To Restore (MTTR)	Average time required to localize a component failure, remove and replace the failed component, and to perform tests to confirm operational readiness of the component.	The sum of the individual times to repair divided by the number of repairs.	Hours.

5.8.6 Service Policies

This standard defines policies as constraints on the allowable actions of an agent or service consumer. The service policies can be described as a part of the JMSDD or, more frequently, in a separate document that contains common policies for a business or organizational domain. The policies can be written in both machine- and human-readable languages (an example of machine-readable policy language is "WS-Policy"; see [38]).

a. Section 4.6 of the JMSDD SHALL provide information about policies that apply to the service.

b. When service policies are presented as a separate document, this policy document SHALL be referenced as prescribed in section 5.6 of this standard.

c. If a referenced policy does not have a persistent Web location (URL), the policy SHALL be included in an Appendix of the JMSDD.

5.8.7 Environmental Constraints

This standard understands environmental constraints as being characteristics of the "super-system", that is, the larger system within which the Java message service operates. Some examples of these constraints are: capacity of existing enterprise network, firewalls, physical computing resources, etc.

a. Section 4.7 of the JMSDD SHALL describe all environmental constraints under which the service is operated and maintained.

b. Section 4.7 of the JMSDD SHALL specify the NEMS environment where the service is to be deployed, indicating whether it is "NAS Operational (NAS-OPS)", "FNTB" or "R&D". See Appendix F for the details for NEMS non-production environments.

c. Section 4.7 of the JMSDD SHALL specify whether the message producer is a NAS or Non-NAS application.

d. Section 4.7 of the JMSDD SHALL specify whether the record type is “live”, “pre-recorded” or “simulated”. See Appendix G for the details for message payload content record types.

5.9 Service Interface

Section 5 of the JMSDD provides detailed information about the types and content of messages that the service exchanges, message exchange models that the service deploys, and any conditions implied by these messages. Note: from the Java perspective, the interface is a programmatic construct that declares a set of methods that other Java classes may implement. However, this standard uses the term *service interface* in a broader sense as being “the means by which the underlying capabilities of a service are accessed”. [SWIM CV]

This standard also asserts that an interface of the service described in the context of a JMSDD is always defined in the JMS API specification [Java Spec].

a. Section 5.1 of the JMSDD SHALL specify the version of the JMS specification used by the service.

b. The specification used by the service SHALL be described as prescribed in section 5.6 of this standard in the Applicable Document section of the JMSDD.

5.9.1 Interfaces

Section 5.1 of the JMSDD describes the interface(s) deployed by the service.

a. Section 5.1 of the JMSDD SHALL present information about the interface being offered by the service.

b. The information SHALL include a name that uniquely identifies the interface throughout the JMSDD.

c. Section 5.1 of the JMSDD SHALL specify the messaging model deployed by the service indicating whether it is “point-to-point (PTP)” or “publish/subscribe”.

It is possible, although rarely recommended, that a service may expose more than one interface.

d. When a service exposes multiple interfaces, each interface SHALL be described in section 5.1 of the JMSDD in accordance with requirements (a) through (c) above.

5.9.2 Operations

In the context of this standard, an *operation* is understood to be a single logical transmission between JMS clients. An operation identifies the sequence and cardinality of messages sent and/or received as well as the client they are logically sent to and/or received from.

In scenarios when a JMS client sends a message and expects to receive a message in return, synchronous request-response messaging can be used. This request-response messaging is often understood to be a subset

of the PTP model. And, although JMS does not explicitly support request-response messaging, its API provides methods that allow for implementation.

a. Section 5.2 of the JMSDD SHALL present information about each operation that the service implements.

b. The information SHALL state if the operation is "synchronous" or "asynchronous".

c. The information SHALL state if the operation is "idempotent" or "non-idempotent".

d. The information SHALL describe the precondition, i.e., the state or condition that should be true before the operation can proceed. A typical example is, "the user has logged in and been authorized".

e. The information SHALL describe the input, i.e., the message that initiates interaction.

f. The information SHALL describe the output, i.e., the data that is produced in response to a request.

g. The information SHALL describe the effect, i.e., the state or condition that exists after the operation is completed (assuming that no error has occurred). For example, "requested map was generated and returned to the message sender".

h. The information SHALL describe the fault(s), i.e., the information that is produced in response to conditions that result in operation failure, including the name of the corresponding fault message described in section 5.4 of the JMSDD.

i. The information SHOULD include a diagram that shows how and in what order messages are exchanged within the context of the operation. Using Unified Modeling Language (UML) diagrams is a RECOMMENDED method for concisely describing concepts without implying a specific technology.

j. If the service does not deploy request-response messaging, the items called for in requirements (a) through (i) MAY be omitted and the phrase, "*This service does not deploy request-response messaging*" inserted in section 5.2 of the JMSDD.

5.9.2.1 Processing Considerations

For the purpose of this standard, *processing* is defined as a set of algorithms, calculations, or business rules that operate on input data in order to produce the required output or to produce a change of internal state. Such actions might be (but are not limited to): transformations of a message by applying algorithms or business rules, or compression or filtering of transmitted data. The processing may be performed not only by a message producer but can also be delegated to a JMS provider (NEMS). In this case, section 5.2.1 of JMSDD can also include, when known, a description of the processing delegated to NEMS.

a. The information SHALL include a description of the processing that takes place within the operation.

b. If all or part of the processing is implemented on NEMS, the information SHOULD include a description of this processing as well.

c. If the processing that takes place on NEMS is described in some other document, this document SHALL be referenced in the Applicable Documents section as prescribed in section 5.6 of this standard.

5.9.3 Messages

Section 5.3 of the JMSDD lists and describes all of the messages exchanged between the JMS clients.

a. Section 5.3 of the JMSDD SHALL list all messages produced or published by the service.

b. Section 5.3.1 of the JMSDD SHALL describe the first message.

c. Every message SHALL be assigned a name that uniquely identifies it throughout the JMSDD.

d. If there are multiple messages, they SHALL be described in consecutive additional sections with headings numbered 5.3.2 (or 5.3.n, as appropriate).

e. The heading of the JMSDD section describing each message SHALL include the name that is used throughout the JMSDD to refer to that message.

f. The information for each message SHALL include the name that uniquely identifies the message throughout the JMSDD. (Note: this is the same name that appears in the heading of the JMSDD section describing the message.)

g. When a message ID is assigned by the message producer, the information SHOULD include the message ID.

h. The information SHOULD include identification of the topic or queue to which the producer sends the message. (Note: identifying the message's destination could be very valuable for those JMS clients that consume messages from more than one topic or queue.)

i. The information SHALL include the message delivery mode, indicating whether it is "persistent" or "nonpersistent". See Appendix E for details.

j. The information SHALL include the message body (payload) type.

k. The single value representing the message body type SHALL be selected from the Table IV.

Table . JMS message body types

Message body type	*Description*
Text	A type of the message body that contains a text, e.g., XML.
Stream	A type of the message body that contains a stream of primitive values that are written and read sequentially.

Map	A type of the message body that contains a set of name-value pairs, where names are strings, and values are Java primitives.
Object	A type of the message body that contains a serialized Java object.
Byte	A type of the message body that contains an array of primitive bytes.

l. When a message priority is assigned by the message producer, the information SHOULD include the message priority, indicated as a numeral ranging from 0 to 9. (Note: priorities 0–4 are usually categorized as “normal” priorities and 5–9 as “expedited” priorities.)

5.9.3.1 Application-Specific Message Properties

Each JMS message contains a built-in facility for supporting message producer-defined properties. Properties provide an efficient mechanism for supporting application-specific message filtering and routing.

a. Section 5.3.1 of the JMSDD SHALL list all properties defined by the message producer.

b. The information for each property SHALL include a name of the property.

c. The information for each property SHALL include a description of the property.

d. The information for each property SHALL include a list of permissible values and a description for each value.

Appendix H contains an example of a producer-defined property.

5.9.4 Exceptions Handling

Section 5.4 of the JMSDD lists and describes all of the faults that are generated in response to conditions that result in failure of an operation or set of operations.

a. Section 5.4 of the JMSDD SHALL list all custom exceptions generated by the message producer.

b. Section 5.4 of the JMSDD SHALL present information about each exception generated by the message producer.

c. The information SHALL include any error code assigned to the exception by the message producer.

d. The information SHALL include any error message that is included as a part of the exception.

e. The information SHALL include a plain language description of the reason for the exception.

5.9.5 Data

Section 5.5 of the JMSDD provides detailed information (i.e., metadata) about the data that constitute a message's body or payload. For the purpose of this standard, two categories of data have been identified: *structured* (e.g., XML documents) and *unstructured* (e.g., images, binary-encoded documents). Among the SWIM-enabled systems, exchange of structured data, serialized as an XML document, is by far the most popular solution. For this reason, this standard focuses on structured data (requirements (**a**) through (**t**)), but it also addresses unstructured data (requirements (**u**) and (**v**)).

a. Section 5.5 of the JMSDD SHALL list all data elements that are part of the message payload.

b. Section 5.5 of the JMSDD SHALL present information about each element listed.

c. The information SHALL include a namespace for the element. Note: if all elements in the list are defined in the same namespace, the namespace can be indicated once for the whole list. See FAA-STD-063[7] for establishing a namespace.

d. The information SHALL include the name of the element.

e. For elements that are registered in the FAA Data Registry (FDR), the information SHALL include the FDR registration identifier. Note: See FAA-STD-060 [6] for information about registering data elements in the FDR.

f. For elements that are registered in the FDR, the information MAY omit items called for in requirements (**g**) through (**q**).

g. The information SHALL include a plain language definition of the element. See section 6.2 of FAA-STD-064 [8] to help ensure that the definition is as informative and understandable as possible.

h. The information SHALL include a description of the element's permissible values in one of the following forms: a range of numbers, a list of individual values, a reference to a source that lists the values (e.g., "FAA Order 7350.7 Location Identifiers"), or a textual description (e.g., "Not Applicable").

i. For elements whose values represent codes, the information SHALL include the meanings of the codes (e.g., "BR = Mist, VA = Volcanic Ash, DU = Widespread Dust, etc.").

j. For elements whose values represent quantitative measures, the information SHALL include the unit of measure (e.g., feet, kilograms, degrees Fahrenheit, dollars).

k. The information SHALL include the element's datatype.

l. For primitive elements, i.e., elements that are not composed of other elements, datatype SHOULD be denoted using the typing system defined in section 3.2 of the W3C XML Schema Part 2: Datatypes specification [40].

m. If another typing system is used, the information SHALL include an explanation of the system or a reference to a source that describes the system. NOTE: if all datatypes are denoted using this system, the explanation or reference may be made once for the whole element list.

n. The information MAY include the element's maximum length together with units of length (e.g., characters, bytes, etc.), if applicable.

o. If an element is to be rendered in a special format, the information SHOULD include a format string.

p. Regular expressions as defined in Appendix F of the W3C XML Schema Datatypes specification [40] are RECOMMENDED for use in describing format strings.

q. If another method is used to express format strings, the information SHALL include an explanation of the method or a reference to a source that describes the method. NOTE: if the method is used to express all format strings, the explanation or reference may be made once for the whole element list.

r. The information SHALL include the obligation ("Required" or "Optional") of the element, i.e., whether the element is required or optional in the context of its underlying information model.

s. The information SHALL include the multiplicity (occurrence) of the element in the context of its underlying information model (e.g., 0, 1, ..., unbounded).

t. Section 5.5 of the JMSDD SHALL include a diagram, or a persistent Web location (URL) of a document or artifact containing such a diagram, that depicts a conceptual or logical model of the data elements listed.

u. If the data that appears in messages is unstructured, section 5.5 of the JMSDD SHALL provide the type, format, and plain language description of the data.

v. If the data that appears in messages is unstructured, section 5.5 of the JMSDD SHALL refer to section 6.1.1.1 (or 6.1.n.1, as appropriate) of the JMSDD that describes how the data should be encoded (see also section 5.10.1.1 of this standard).

5.9.5.1 Referencing External Data Documents

Sometimes a document that describes the data exchanged by JMS clients is produced separately, often by an organization other than the organization responsible for developing the JMSDD. For the purpose of this standard, such a document will be referred to as an External Data Document (EDD). Usually an EDD is developed for use by multiple services and not just for the service described in the JMSDD. If such an EDD exists, it would seem logical and efficient to provide a reference to the EDD instead of copying its content into section 5.5 of the JMSDD. If an EDD is to be used as a substitute for the content prescribed above for section 5.5 of the JMSDD, the EDD must comply with following requirements:

a. The EDD SHALL conform to the requirements (**a**) through (**v**) specified in section 5.9.5 of this standard.

b. The EDD SHALL maintain a versioning policy.

c. The EDD SHALL have a persistent Web location (URL).

d. The EDD specified in section 5.5 of the JMSDD SHALL be documented as prescribed in the Applicable Documents section 5.6 of this standard.

5.10 Service Implementation

The JMS as an abstract specification does not specify any "on-wire" protocols, and therefore any implementation of JMS providers and clients is always constrained or enabled by a vendor-provided set of protocols. Section 6 of a JMSDD presents information about the software product that is used for developing and operating the service, as well as details about how the service can be accessed, i.e., information about data, transport, and message protocols as well as network address.

a. Section 6 of the JMSDD SHALL present information about the implementation of the service.

b. The information SHALL include the name of the product being deployed as a JMS provider.

c. The information SHALL include the version of the product being deployed as a JMS provider.

5.10.1 Bindings

For the purpose of this standard, a binding is understood to be a named collection of protocols which are used in the course of an execution of the service.

The binding concept is depicted in Figure V.

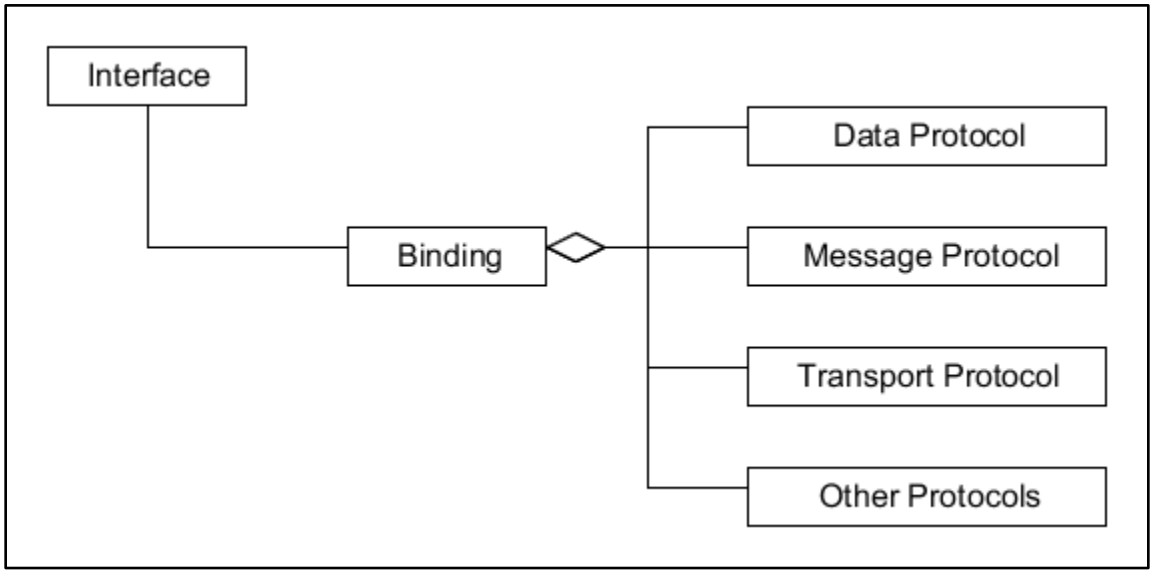

FIGURE . Binding - conceptual diagram

a. Section 6.1 of the JMSDD SHALL list all bindings implemented by the service.

b. Section 6.1.1 of the JMSDD SHALL describe the first binding.

c. If there are multiple bindings, they SHALL be described in consecutive additional JMSDD sections with headings numbered 6.1.2 (or 6.1.n, as appropriate).

d. The heading of the JMSDD section describing each binding SHALL include the name that is used throughout the JMSDD to refer to that binding.

e. Each binding description SHALL include the name of the operation, and the name of an interface associated with this operation, that deploys this binding.

f. The name of the operation and/or associated interface SHALL be consistent with the operation and/or interface name established in section 5.2 and/or 5.1 respectively of the JMSDD.

g. Each binding description SHALL include information about each data protocol that the service uses for this binding, as described in section 5.10.1.1 of this standard.

h. Each binding description SHALL include information about the message protocol that the service uses for this binding, as described in section 5.10.1.2 of this standard.

i. Each binding description SHALL include information about the transport protocol that the service uses for this binding, as described in section 5.10.1.3 of this standard.

j. Each binding description SHALL include information about other protocols that the service uses for this binding, as described in section 5.10.1.4 of this standard.

k. When the organization responsible for supporting and/or operating the NEMS provides a formal document that specifies information about the NEMS protocols, that document MAY be referenced in lieu of presenting binding information as prescribed in requirements (a) through (j).

l. The formal document SHALL be documented as prescribed in section 5.6 of this standard.

5.10.1.1 Data Format

A data format is a set of rules governing data serialization and coordination for data exchange among components.

The XML format is presently most often employed for exchanging textual data in JMS implementations. Besides transmitting textual data, an important use for XML is serializing data structures according to domain-specific conceptual models; e.g., the Geography Markup Language (GML) used to serialize information about geographical features or the Aeronautical Information Exchange Model (AIXM) used for transmitting aeronautical information.

a. Section 6.1.1.1 (or 6.1.n.1, as appropriate) of the JMSDD SHALL describe the data format for this binding.

b. Section 6.1.1.1 (or 6.1.n.1, as appropriate) of the JMSDD SHALL include the normative document that regulates the data format.

c. The normative document SHALL be described as prescribed in section 5.6 of this standard.

d. If an accessible reference (e.g., URL) to the normative document is not available, the document itself SHALL be included in an Appendix of the JMSDD.

e. The data protocol described in section 6.1.1.1 (or 6.1.n.1, as appropriate) of the JMSDD SHALL be compatible with the data defined in section 5.5 ("Data") of the JMSDD.

5.10.1.2 Message Protocol

In an enterprise messaging environment, the communication and interaction between components is performed by exchanging messages of predefined content. A message protocol is a formal set of rules and conventions governing procedure calls and responses among communicating SOA components.

a. Section 6.1.1.2 (or 6.1.n.2, as appropriate) of the JMSDD SHALL describe the message protocol for this binding.

b. Section 6.1.1.2 (or 6.1.n.2, as appropriate) of the JMSDD SHALL include the normative document that establishes the message protocol.

c. The normative document SHALL be described as prescribed in section 5.6 of this standard.

d. If an accessible reference (e.g., URL) to the normative document is not available, the document itself SHALL be included in an Appendix of the JMSDD.

5.10.1.3 Transport Protocol

A transport protocol is a formal set of rules governing message transmission and port handling among communicating SOA components. JMS as an abstract specification does not specify any transport protocol. Vendors supporting JMS implementations, like WebLogic JMS or ActiveMQ, usually offer multiple transports. In ActiveMQ, for example, the transport protocol can be TCP, SSL or HTTP.

Note: this standard asserts that all JMS clients require a trusted network connection and use TCP/IP as an underlying transport protocol.

a. Section 6.1.1.3 (or 6.1.n.3, as appropriate) of the JMSDD SHALL describe the transport protocol for this binding.

b. Section 6.1.1.3 (or 6.1.n.3, as appropriate) of the JMSDD SHALL include the normative document that establishes the transport protocol.

c. The normative document SHALL be described as prescribed in section 5.6 of this standard.

d. If an accessible reference (e.g., URL) to the normative document is not available, the document itself SHALL be included in an Appendix of the JMSDD.

5.10.1.4 Other Protocols

Some modern protocols may combine data definitions with messaging conventions or messaging and transport governing conventions and cannot be unambiguously classified as strictly a data, message or transport protocol to be described in section 6.1.1.1, 6.1.1.2 or 6.1.1.3 of the JMSDD. This section of the standard describes requirements for such protocols.

a. Section 6.1.1.4 (or 6.1.n.4, as appropriate) of the JMSDD SHALL describe any other protocols that the service uses for this binding and that cannot be clearly identified as a data protocol, a transport protocol, or a message protocol.

b. Section 6.1.1.4 (or 6.1.n.4, as appropriate) of the JMSDD SHALL include the normative document that establishes each described protocol.

c. The normative document SHALL be described as prescribed in section 5.6 of this standard.

d. If an accessible reference (e.g., URL) to the normative document is not available, the document itself SHALL be included in an Appendix of the JMSDD.

5.10.2 End Points

In the context of this standard, an end point is understood to be an association between a fully-specified binding, as described in section 5.10.1 of this standard, and "a physical point at which a service may be accessed" [12], i.e., a network address, which can be a fully qualified URI or a JNDI reference.

Note: in the context of JMS, end points are not directly accessible to each other: a message producer does not send messages to a message consumer end point, it delivers messages to an abstract end point (a queue or a topic); by the same token, a message consumer does not fetch messages from a message producer end point, it obtains the messages from one or more end points (queues or topics) internal to the JMS provider.

The end point concept is depicted in Figure VI.

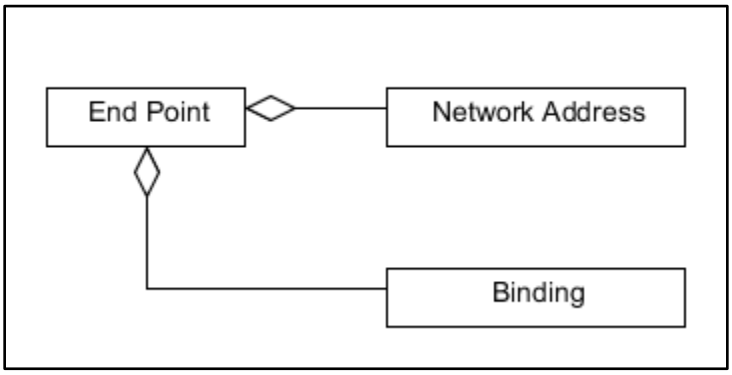

FIGURE . End point - conceptual diagram

a. Section 6.2 of the JMSDD SHALL list all end points implemented by the service.

b. Section 6.2.1 of the JMSDD SHALL describe the first end point.

c. If there are multiple end points, they SHALL be described in consecutive additional JMSDD sections with headings numbered 6.2.2 (or 6.2.n, as appropriate).

d. The heading of the JMSDD section describing each end point SHALL include the name that will be used throughout the JMSDD to refer to that end point.

e. Each end point description SHALL include the name of the binding, and the network address.

f. The name of the binding SHALL be consistent with the binding name established in section 6.1 of the JMSDD.

APPENDIXES

Appendix A. Example of a JMSDD Cover Page

FAA-X-XXX
Revision A
September 4, 2012

U.S. Department of
Transportation
**Federal Aviation
Administration**

Java Messaging Service Description Document
Special Activity Airspace Management Service (SAAMS)

Appendix B. Example of a JMSDD Approval Signature Page

FAA-X-XXX
Revision A
September 4, 2012

Java Messaging Service Description Document
Special Activity Airspace Management Service (SAAMS)

Approval Signatures

Name	Organi-zation	Signature	Date Signed

ii

Appendix C. Example of a JMSDD Revision Record Page

FAA-X-XXX
Revision A
September 4, 2012

Revision Record

Revision Letter	Description	Revision Date	Entered By

iii

Appendix D. Examples of Quality of Service (QoS) Parameters

The table below provides examples of QoS parameters that are relevant to a message producer. JMSDD developers may reuse these parameters or provide their own, as well as their own values or range of values.

QoS Parameter Name	*Value or Range of Values*	*Definition*	*Method*	*Unit of Measure*
Accuracy	250	Number of errors produced by the service over a period of time.	Simple count. Measurements are taken daily and apply to the preceding 24-hour period.	Whole positive number
Capacity	25 per minute	Number of service requests that the service can accommodate within a given time period.	Simple count.	Whole positive number, per period of time.
Response Time	10	Maximum time required to complete a service request.	Measured from the time the message producer receives the request to the time the message producer transmits the response.	Seconds.

Appendix E. JMS Message Delivery Modes

The table below provides descriptions and explanations of JMS message delivery modes, as described in section 5.9.3 of this standard. The source of these descriptions is Java Message Service Specification Version 1.1: Sun Microsystems, Inc.: April 12, 2002[43]

Delivery Mode	***Description***
Persistent	The "persistent" mode instructs the JMS provider to take extra care to insure the message is not lost in transit due to a JMS provider failure. A JMS provider must deliver a *persistent* message *once-and-only-once*. This means a JMS provider failure must not cause it to be lost, and it must not deliver it twice.
Non-persistent	A JMS provider must deliver a "non-persistent" message *at-most-once*. This means that it may lose the message, but it must not deliver it twice.

Appendix F. NEMS Non-Production Environments

The table below provides descriptions for NEMS non-production environments as described in section 5.8.7 of this standard.

Environment Name	***Description***
FAA National Test Bed (FNTB)	The environment for operations testing of services prior to approval for deployment to the NEMS.
Research and Development (R&D)	The environment for prototyping, proof-of-concept, and early development integration of services.

Appendix G. Message Payload Content - Record Types

The table below provides descriptions for message payload content record type as described in section 5.8.7 of this standard.

Record Type	***Description***
Live	Operations data permissible only on the NEMS
Pre-recorded	Old data previously recorded on the NEMS being reused on the non-production environments for testing purposes.
Simulated	Mock data created in the non-production environments for scenario simulation testing purposes

Appendix H. Example of Producer-Defined Message Properties

The table below provides an example of a producer-defined message property as described in section 5.9.3.1 of this standard. Note: this example is non-normative.

Name	*Description*	*Permissible Values*
FlightSensitivityType	The property, of the type String, that indicates the level of discretionary access to the flight information contained in the message.	**"military"** - a flight is considered to be a military flight if it is operated by any arm of the military of the United States or any other government. **"sensitive"** - a flight is considered to be sensitive if the flight operator or owner requests the FAA to consider it to be sensitive.